# Summary of “preprocessing” steps to obtain accurate Dexcom timestamps

**Dexcom** timestamp variables:

* *Transmitter Time*: the counter of the transmitter in seconds, it starts counting when it is activated (?)
* *“Local time”*: the local timestamp according to the device that they connect to (later referred to as the “dexcom local time”) - the device that they connect to can either be a phone or the dexcom receiver.

**TrainingPeaks** files from the training sessions contain UTC timestamps and locations. From this we can make a list of time ranges when they were in certain countries.

To match TrainingPeaks with Dexcom, we need both of them to be in *UTC.* (We also need this as an intermediate step to obtain correct Dexcom timestamps.)

1. From the TrainingPeaks data, create a list of locations/timezones for each athlete on each date of which we know they cycled
   1. For each file of a training session, we obtain the location, and the timezone that is in the header of the file.
   2. Missing location or timezones: If the location or timezone for all training sessions two days before and two days after a given training session with missing timezone or location is the same, we assume it has this timezone or location.
   3. If the timezone and location do not match, we assume that the location is most likely the correct entry here, and we replace the timezone with the timezone associated with that location.
2. From the Dexcom data, create a list of timezone *changes* for each athlete
   1. Select only EGV measurements
   2. Sort by {rider, transmitter ID (in chronological order), transmitter time}
   3. Define:
      1. *transmitter\_time\_diff*: the difference in transmitter time between two sequential readings
      2. *local\_timestamp\_diff*: the difference in “dexcom local time” between two sequential readings
      3. *timediff* = *local\_timestamp\_diff - transmitter\_time\_diff* : the difference in “dexcom local time” between two sequential readings where we correct for any *censoring* (i.e. time period with no CGM readings)
   4. Identify *gaps* (positive change in timezone (e.g. from UTC+2 to UTC+3)) and *duplicates* (negative change in timezone (e.g. from UTC+2 to UTC+1))
      1. GAP: when there is a *timediff* between two sequential readings that is larger than 5 min (or 55 min, does not matter) (and no change in rider or transmitter)
      2. DUP: when there is a *timediff* between two sequential readings that is smaller than -5 min (or -55 min, does not matter) (and no change in rider or transmitter
   5. Identify potential timezone changes that cannot be observed directly from the dexcom data with *gaps* and *duplicates*, for example any changes that they made in timezone in between two transmitters. This usually happens when they were not doing any CGM readings for a few months, then had to change the transmitter when it expired. If they were using the dexcom receiver instead of a phone, it does not change from winter to summer time automatically (because there is no internet connection), and therefore they might change the timezone of the receiver manually, before putting on the new transmitter. This is a change that we would not be able to see directly from the dexcom data through these *gaps* and *duplicates.*
3. Cross-match the timezone *changes* from Dexcom with the list of timezones from TrainingPeaks. Note that we have only *changes* in timezone from the Dexcom data, and not the actual timezones. By looking at the first and last date they were in a given timezone from TrainingPeaks, we cross-match any change in dexcom that happened between the last date of timezone A and the first date of timezone B from the TrainingPeaks table. Unfortunately this only covers 20% of all timezone changes for the following reasons:
   1. They often changed the timezone of the device that they connected to (either phone or receiver) manually. This was done later than the first day that they cycled again. (Not only just one day, but can be a few weeks.)
   2. Many timezone changes happen because they switch between a device that is in timezone A and a device that is in timezone B (so this is not actually due to travelling)
   3. TrainingPeaks does not capture any timezone changes that happen when they are not on the bike (sometimes they do intermediate travels)
   4. It can also happen that they do not change the timezone of their device when they are travelling
   5. They can make mistakes when they reset the timezone on their phone.

Many changes therefore had to be identified manually. We calculate the timezone that we should obtain with the timezone change, to check if it matches with what we see in the TrainingPeaks data. Now we have obtained a table with entries {athlete, timezone, local\_timestamp\_min, local\_timestamp\_max} which is the timezones for each athlete in the dexcom data.

1. With the list of timezones in the dexcom data, we calculate the **UTC timestamps in the dexcom data**. (Details: the local\_timestamp\_min and local\_timestamp\_max are connected to an index in the dexcom data. If we would simply say: anything in between these local timestamp ranges connects to a certain timezone, we would still include the duplicate local timestamp due to travelling, and these would then not be correctly transformed to UTC.)

To identify nocturnal hypoglycaemia (as well as any glucose ranges during any part of the day), we need Dexcom data in correct local time.

Note that the timezone changes in the dexcom data that we observe do not match with the actual local time, they are often ~1 week off, as the athletes do not change the timezone of their device directly after travelling (or they do not change it at all).

1. Create a list of races and camps for each athlete
   1. Find the dates of cycling races that each athlete participated in at procyclingstats.com
   2. Look into the trainingpeaks calendar online to find any race or camp entries in their calendar
   3. Look into the calendar excel sheet Fede provided to identify any race or camp entries
2. Check with the TrainingPeaks cycling data to see if they were at any of the locations identified in the previous step (if not, remove).
3. For each timezone observed in the dexcom data, check manually if this is actually correct
   1. Match each timezone in dexcom with a timezone in trainingpeaks, and check if the change between timezones in dexcom actually corresponds with what we see in trainingpeaks (e.g. was it in time or too late?)
   2. Identify whether a rider changed the timezone on his device manually or automatically (after travelling or summer/winter time)
      1. If he used a dexcom receiver instead of a phone → manually
      2. If the timezone changes happened too late → manually
      3. If the timezone changes happened always in time → automatically
      4. If there is any intermediate timezone change registered (e.g. a stop-over) → potentially automatically

If a timezone change happened too early (this can also happen), there are two explanations: either there is something wrong with the location or date observed in trainingpeaks (this very rarely happens with the garmin devices), or they were very eager to change their timezone manually before their flight.

* 1. For all riders that changed the timezone on their device manually, we find out manually what the actual date is that they changed to this timezone.
     1. We take the time window in TrainingPeaks associated with this timezone.
     2. We go online to the trainingpeaks calendar to see if there is any travelling entry in their calendar, as well as whether there is any day on which they did not cycle within that time window.
     3. We check if there is a censoring gap in the dexcom data within that time window that could cover the length of travel to that timezone. What could happen is that they take off the transmitter while travelling, because it is advised not to go with the transmitter through an X-ray. (Could it also be possible that they did not record for a while because the phone was in airplane mode?)
     4. If we find a censoring gap that is big enough for the travelling, and that also took place *before* they changed the timezone on the dexcom manually, we say that this is the time that they travelled. If we do not find a censoring gap that is big enough, we approximate the arrival time by taking 00:00 at old local time on the day that we identified in the trainingpeaks calendar.